GUJARAT TECHNOLOGICAL UNIVERSITY

Program Name: Bachelor of Computer Applications

Level: UG

Course / Subject Code: BC03001061

Course / Subject Name: Version Controlling

w. e. f. Academic Year:	2025-26
Semester:	3
Category of the Course:	Ability Enhancement Courses

Prerequisite:	Basic computer literacy, fundamentals of programming, and basic command line usage.
Rationale:	In the modern software development landscape, version control systems (VCS) are essential tools for effective and collaborative coding. Among them, Git has emerged as the industry standard, while GitHub is widely adopted for cloud-based source code management. However, many students and early-career developers lack formal exposure to these tools, which creates a significant skill gap as they transition into internships, collaborative projects, or professional environments. This 14-session, hands-on practical course is designed to bridge that gap, enabling learners to master version control practices from basic to intermediate levels using Git and GitHub. **Course Objectives** The broad course objectives are as – • Introduce the fundamentals of version control and Git architecture. • Provide hands-on experience with local and remote Git workflows. • Develop collaborative coding skills using GitHub features like branching, pull requests, and issue tracking. • Promote best practices in software project documentation, commit history, and conflict resolution. • Empower students to manage individual and team projects with professional standards. **Pedagogy** This course is structured around active, experiential learning through: • Real-world exercises and step-by-step tutorials • Assignments and peer collaboration via GitHub Classroom • Project-based learning that culminates in a team-driven software development task • Regular assessment through practical sessions, quizzes, and code reviews The combination of theoretical grounding and tool-based practice ensures that learners not only understand the “why” behind version control but can also confidently apply the “how” in real projects. **Mastery of version control tools is:** Essential for software engineering, data science, web development, and open-source contributions.

	A core skill demanded by employers across industries. A foundation for more advanced practices like Continuous Integration/Deployment (CI/CD), DevOps, and Agile workflows. Moreover, it enhances students' project organization, collaboration, debugging, and documentation capabilities—traits highly valued in both academia and the industry.

Course Outcome:

After Completion of the Course, students will be able to:

No	Course Outcomes	RBT Level
01	Discuss the importance of version control in software development.	UN
02	Use appropriate Git commands.	AP
03	Manage repositories using GitHub.	AP
04	Collaborate on projects using Git workflows (feature branching, pull requests, etc.).	AP
05	Use Git in real-world projects with best practices.	AP

**Revised Bloom's Taxonomy (RBT)*

Teaching and Examination Scheme:

Teaching Scheme (in Hours)			Total Credits L+T+ (PR/2)	Assessment Pattern and Marks				Total Marks
				Theory		Tutorial / Practical		
L	T	PR	C	ESE (E)	PA / CA (M)	PA/CA (I)	ESE (V)	
0	0	4	2	0	0	50	50	100

Course Content:

Unit No.	Content	No. of Hours

1	**Session 1: Introduction to Version Control**
	• Concepts: Version control systems, centralized vs distributed VCS
	• Tools: Overview of Git, GitHub, GitLab, Bitbucket
	• Setup: Installing Git, configuring Git (username, email)
	• Hands-on:
	  ○ Creating a project folder
	  ○ Basic Git commands (git init, git status, git config)
	• Practice: Create and track a simple text file
		4

2	**Session 2: Git Basics - Adding and Committing** • Concepts: Working directory, staging area, local repository • Commands: git add, git commit, git log • Hands-on:	4

	○ Create a sample project ○ Track file changes, make multiple commits ○ Viewing history with git log, git show • Practice: Project logbook tracking	
3	**Session 3: Working with GitHub** • Concepts: Remote repositories, SSH vs HTTPS • Hands-on: ○ Create a GitHub account ○ Create and clone repositories ○ Connect local repo to remote (git remote, git push) • Practice: Push a simple project to GitHub	4
4	**Session 4: Cloning and Collaboration** • Concepts: Cloning, pulling changes • Commands: git clone, git pull, git fetch • Hands-on: ○ Clone a GitHub repository ○ Make changes and push updates • Practice: Syncing team projects	4
5	**Session 5: Branching in Git** • Concepts: Why use branches, feature branching • Commands: git branch, git checkout, git switch, git merge • Hands-on: ○ Create and switch branches ○ Merge branches and handle fast-forward • Practice: Create a feature and merge it	4
6	**Session 6: Conflict Resolution** • Concepts: Merge conflicts and resolution strategies • Commands: git diff, git mergetool • Hands-on: ○ Create conflict scenarios ○ Resolve conflicts using CLI and GUI tools • Practice: Conflict resolution exercise	4
7	**Session 7: GitHub Collaboration – Pull Requests** • Concepts: Forking, Pull Requests (PRs), Code Review • Hands-on: ○ Fork a repo, create a branch, push changes ○ Create a PR, review and merge it • Practice: Mini collaborative project via GitHub PR	4

8	**Session 8: Undoing Changes**
	• Concepts: reset, revert, checkout, restore
	• Hands-on:
	  ○ Undo local changes
	  ○ Revert commits
	  ○ Discard staged files
	• Practice: Recovery scenarios
		4

9	**Session 9: Git Tags and Releases**
	• Concepts: Tagging for versioning, annotated vs lightweight
	• Hands-on:
	  ○ Create tags, list, and delete tags
	  ○ Push tags to the remote
	  ○ Use GitHub releases
	• Practice: Tag versions in a project
		4

10	**Session 10: Git Log and History Navigation**
	• Concepts: Viewing and interpreting commit history
	• Commands: git log, gitk, git blame, git bisect
	• Hands-on:
	  ○ Browse project history
	  ○ Use blame to track changes
	• Practice: Debugging using commit history
		4

11	**Session 11: .gitignore and Git Attributes**
	• Concepts: Ignoring files and customizing behavior
	• Hands-on:
	  ○ Create .gitignore for common environments
	  ○ Explore .gitattributes for merge/diff strategies
	• Practice: Create .gitignore for a Python or Node.js project
		4

12	**Session 12: Working with GUI Tools**
	• Tools: GitHub Desktop, GitKraken, VS Code Git integration
	• Hands-on:
	  ○ Visualize repo with GUI
	  ○ Perform commits, merges, and pull requests with GUI
	• Practice: Repeat earlier exercises using GUI
		4

13	**Session 13 ,14, and 14: Project Work – Team Collaboration**
	• Task:
	  ○ Teams of 2–4 students collaborate on a mini software project
	  ○ Use GitHub for project tracking and collaboration
	  ○ Use branches, PRs, and issue tracking
	• Deliverable:
	  ○ Final codebase and Git history
	  ○ Documentation of the workflow used
		8

14	**Session 15: Assessment and Best Practices**
	• Review: Key concepts, common pitfalls
	• Best Practices:
	  ○ Commit message conventions
	  ○ Branch naming strategies
	  ○ Writing a clean history with rebase
	• Assessment:
	  ○ Practical test
	  ○ Viva on workflows and GitHub usage
	  ○ Evaluation of project work
		4

	Total	**60**

Suggested Specification Table with Marks (Theory):

Distribution of Theory Marks (in %)					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
10	20	70	-	-	-

Where R: Remember; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (as per Revised Bloom's Taxonomy)

References/Suggested Learning Resources:

(a) Books:

1. Git Book https://git-scm.com/book/en/v2, GitHub Learning Lab

(b) Tools:

1. Git: https://git-scm.com/
2. GitHub: https://github.com/
3. GitHub Desktop: https://desktop.github.com/

(c) Resources:

1. https://spoken-tutorial.org/tutorial-search/?search_foss=Git&search_language=English

CO- PO Mapping:

Semester 3	**Course Name : Version Controlling**										
	POs										
Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	1	-	-	-	1	-	-	-	-	-	-
CO2	1	-	-	-	3	-	-	3	-	-	2
CO3	1	-	-	-	3	-	-	3	-	-	2
CO4	1	-	-	-	3	-	-	3	-	-	2
CO5	1	-	-	-	3	-	-	3	-	-	2

*Legend: '**3**' for high, '**2**' for medium, '**1**' for low and '**-**' for no correlation of each CO with PO.*

Note: The CO-PO mapping is indicative; the institute/faculty member can change as required.

* * * * * * *